



1/666/666

STATE OF ISRAEL

REC'D 29 DEC 1999

WIPO PCT

This is to certify that  
annexed hereto is a true  
copy of the documents as  
originally deposited with  
the patent application  
particulars of which are  
specified on the first page  
of the annex.

זאת לתעודה כי  
רוצפים בזיה העתקים  
נכונים של המסמכים  
שהופקרו לכתחילה  
עם הבקשה לפטנט  
לפי הפרטים הרשומים  
בעמוד הראשון של  
הנספח.

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

12-12-1999

This \_\_\_\_\_ day of \_\_\_\_\_

רשות הפטנטים  
Registrar of Patents

נחתasar

לשימוש הלשכה  
For Office Use

מספר: Number	127437
תאריך: Date	17-12-1998
הקודם/נדחת Ante/Post-dates	

1967 חוק הפטנטים, התשכ"ז  
PATENTS LAW, 5727-1967

בקשה לפטנט  
Application for Patent

C:32682

אני, (שם המבקש, מעת -- ולגבי נוף מאוגנד -- מקום התאגדותו)  
I (Name and address of applicant, and, in case of body corporate-place of incorporation)

RDC COMMUNICATIONS LTD.

1 Hamelacha Street  
Northern Industrial Zone  
Lod 71293

(An Israeli company)

אור. ד. סי. תקשורת בע"מ  
רחוב המלאכה 1  
אזור תעשייה צפוני  
לוד 71293

(חברה ישראלית)

By Law  
Owner, by virtue of

בעל אמצעה מכח הדיין  
of an invention, the title of which is:

התקן ושיטות לאספקת איכות שירות בתוך מערכת עם לופ לוקלי אלחוטי

(בעברית)  
(Hebrew)

APPARATUS AND METHODS FOR PROVIDING QUALITY OF SERVICE WITHIN A  
WIRELESS LOCAL LOOP SYSTEM

(באנגלית)  
(English)

hereby apply for a patent to be granted to me in respect thereof

מבקש בזאת כי ינתן לי עליה פטנט

Application for Division	* בקשה חילוקה Application for Division	* בקשה פטנט מוסף - Application for Patent of Addition	דרישה דין קדימה Priority Claim		
			מספר/סימן Number/Mark	תאריך Date	מדינת האינט Convention Country
מבקשת פטנט from Application	No. _____ מס. _____ dated _____ מיום _____	* לבקשת/לפטנט to Patent/Appl. No. _____ מס. _____ dated _____ מיום _____			
<p>* יפי כה: כלל/מיוחד - רצף בזיה / עוד יונש P.O.A.: general / individual - attached / to be filed later - הונש בעניין _____ filed in case _____</p> <p>המע למסירת הודעות ומסמכים בישראל Address for Service in Israel <u>Sanford T. Colb &amp; Co.</u> <u>P.O.B. 2273</u> <u>Rehovot 76122</u></p>					
<p>חותמת המבקש Signature of Applicant</p> <p>For the Applicant,</p> <p><u>Sanford T. Colb &amp; Co.</u> C:32682</p>			<p>1998      DECEMBER שנת      בחודש of the year      of 7      This יום This</p> <p>לשימוש הלשכה For Office Use</p>		

טופס זה, כשהוא מוטבע בחותם לשכת הפטנטים ומוסלם בספר ובתאריך ההגשתה, הינו אישור להגשת הבקשה שפטוטיה רשותים לעיל.  
This form, impressed with the Seal of the Patent Office and indicating the number and date of filing, certifies the filing of the application,  
the particulars of which are set out above.

\* Delete whatever is inapplicable

התקן ושיטות לאספקת איכות שירות בתוך מערכת עם לוֹפְּ לְוקְלי אַלְחוּטִי  
APPARATUS AND METHODS FOR PROVIDING QUALITY OF SERVICE  
WITHIN A WIRELESS LOCAL LOOP SYSTEM

RDC COMMUNICATIONS LTD.

אר. די. סי. תקשורת בע"מ

C:32682

## APPARATUS AND METHODS FOR PROVIDING QUALITY OF SERVICE WITHIN A WIRELESS LOCAL LOOP SYSTEM

### FIELD OF THE INVENTION

The present invention relates to wireless information transactions.

### BACKGROUND OF THE INVENTION

The state of the art as pertaining to quality of service, wireless local loop systems and the Internet generally is exemplified in the following publications:

- [1] G. Mapp and S. Hodges. QoS-Based Transport.
- [2] J. Crowcroft and P. Oechslin. Differentiated End-to-End Internet Services using a Weighted Proportional Fair Sharing TCP.
- [3] D. K. H. Tan. Rate control and User Behaviour in Communication Networks.

The disclosures of all publications mentioned in the specification and of the publications cited therein are hereby incorporated by reference.

## SUMMARY OF THE INVENTION

The present invention seeks to provide Quality of Service to an information transaction between two peers within a Wireless IP Local Loop (WipLL) system.

**Quality-of-Service (QoS)** is a term that is related to sessions.

A session is defined as an information transaction between two or more peers.

**QoS of a session** is a set of conditions that should be maintained during the information transaction, e.g., required bandwidth (Kbps), transaction latency (delay), tolerance to jitter (variation of delay), tolerance to information loss, etc.

A **network** is a graph that its nodes are peers exchanging information, and its edges are the physical connection media, e.g., copper wiring.

**Congestion** is a temporary information flow block that occurs in heavy loaded networks.

A loaded network presents different behaviors depending on its topology and information load, one particular behavior is congestion. Congestion leads to starvation (spatio-temporal access blocking to physical resources). Starvation in its turn leads to large transaction latencies and cutouts.

**Congestion versus session behavior:**

All sessions present defined timeouts for the information stream delay. Some sessions have defined and constant used bandwidth, e.g., telephone sessions. Other sessions use all available bandwidth - the available channel is sensed by the return path delay, and the information transmission rate is adjusted accordingly (the session uses all available bandwidth, but minimizes system's queues).

If all sessions were bandwidth adjustable, all queues in the system would then become very small. Thus, in an overloading scenario, rate adaptation and session denial could maintain QoS. As "real-world" networks (and webs) are a mixture of sessions, some of which can not be rate adjusted, some presenting dramatically different delay requirements, etc., bandwidth adjustment alone is not enough to prevent congestion and starvation..

The present invention preferably utilizes three simultaneous approaches to provide Quality of Service to an information transaction between two peers within a

WipLL system: Weighted Fair Queuing (WFQ, based on the time-to-live of the different packets in the transmission queue of a node); rate-control; and QoS-scheduling. The present invention particularly utilizes WFQ and QoS-scheduling and the adaptive combination between them.

Furthermore, the WFQ can handle node-queues, but it can not handle the all access system (e.g. star topology) queuing without some introduction of congestion. Thus, Weighted-queuing is performed between the different node-queues in the system. Each queue is represented by a severity grade that the network master allocated.

The WipLL system presents a QoS mechanism that preferably includes some or all of the following three major features:

1. Adaptive network filtration and forwarding agent, responsible for forwarding packets that "belong" to the wireless channel only. Thus, irrelevant data streams are filtered out and do not compete over the air channel.
2. The QoS server is responsible for network and application layers policies execution including analyzing each incoming packet, detecting its session, evaluating channel load, performing flow control operations (such as delaying packets, intervening into the connection layer, etc.) and attaching a QoS header (over the air) to a packet describing the packet's boundary conditions (such as retransmission criteria, TTL, etc.). It is emphasized that TCP (for example) rate control can be carried out in such a way that queues within the access system are kept in constant length. This in turn leads the system into additional minimization of session jitter, and thus enhances performance.
3. Classified queuing for TTL adaptive access latency within the MAC domain, enabling optimal channel bandwidth control (for queued data regimes).

These three features in complementary operation within the system guarantees quality of service within an integrated services system.

There is thus provided in accordance with a preferred embodiment of the present invention a quality of service system including a congestion avoidance subunit

wherein the congestion avoidance unit is operative to perform classified queuing, and a traffic flow control unit.

There is further provided in accordance with another preferred embodiment of the present invention a quality of service server apparatus including a protocol detector, and a connection layer analyzer including a UDP analyzer, a TCP analyzer and an ICMP analyzer.

Further in accordance with a preferred embodiment of the present invention the UDP analyzer includes a rate controlled UDP analyzer.

Still further in accordance with a preferred embodiment of the present invention the TCP analyzer includes a rate controlled TCP analyzer.

Additionally in accordance with a preferred embodiment of the present invention the UDP analyzer performs at least some of the following steps: identifies applications by using its port number, checks whether packet belongs to already open session by comparing port numbers and session's participant's IP addresses, if it is an open session, stamp packet with TTL from applications lookup table, if it is a new session, consult with policy agent to determine whether this session is allowed to initiate, inform MAC on application's covenant, in terms of CS air MAC-address and inform MAC about session end events.

Further in accordance with a preferred embodiment of the present invention the TCP analyzer performs at least some of the following reliability checks, acknowledges receipt of packets, retransmits when dropped packets are detected, re-sequences segments, if necessary, if they arrive out of order, tosses packets if data became corrupt during transmission, discards duplicate segments and maintains flow control to manage a connection's transmission rate.

Still further in accordance with a preferred embodiment of the present invention the congestion avoidance unit is operative to perform classified queuing.

Additionally in accordance with a preferred embodiment of the present invention the rate of TCP transmissions is at least partly controlled by detecting real-time flow speed and then delaying ACKs going back to the transmitter.

Further in accordance with a preferred embodiment of the present invention the rate of TCP transmissions is at least partly controlled by modifying the advertised window size in the packets sent to the transmitter.

Still further in accordance with a preferred embodiment of the present invention the classified queuing performed by the congestion avoidance unit includes assigning packets arriving with a time-to-live stamp to the transmit-queue cluster, to a queue according to their time-to-live indicator.

There is further provided in accordance with another preferred embodiment of the present invention a quality of service system including an adaptive network filtration and forwarding agent, a quality of service server, and a classified queuing mechanism.

Further in accordance with a preferred embodiment of the present invention the agent is operative to forward packets that belong to the wireless channel only while filtering out irrelevant data streams and competing over the air channel.

Still further in accordance with a preferred embodiment of the present invention the quality of service server is operative to execute network and application layers policies including executing at least one of the following: analyzing each incoming packet to detect its session, evaluating channel load, performing flow control operations such as delaying packets and intervening into the connection layer, and attaching a quality of service header to the packet describing the packet's boundary conditions.

Additionally in accordance with a preferred embodiment of the present invention rate control is carried out such that queues within the access system are kept at substantially a constant length, thereby to reduce session jitter.

Further in accordance with a preferred embodiment of the present invention the classified queuing mechanism provides TTL adaptive access latency within the MAC domain, thereby to enable improved channel bandwidth control for queued data regimes.

## BRIEF DESCRIPTION OF THE DRAWINGS AND APPENDICES

The present invention will be understood and appreciated from the following detailed description, taken in conjunction with the drawings and appendices in which:

Fig. 1 is a simplified block diagram of global data flow within a Wireless IP Local Loop (WipLL) system constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 2 is a simplified flowchart illustrating TCP rate control.

Fig. 3 is a simplified illustration of a QoS server controlling data transmission constructed and operative in accordance with a preferred embodiment of the present invention; and

Fig. 4 is a simplified block diagram of data traffic behavior both with and without the control of a QoS server system constructed and operative in accordance with a preferred embodiment of the present invention.

Fig. 5 is a simplified block diagram of the operation of a QoS server system constructed and operative in accordance with a preferred embodiment of the present invention.

Attached herewith is the following appendix which aids in the understanding and appreciation of one preferred embodiment of the invention shown and described herein:

Appendix A is a listing of a method for performing Queue Weighting and Specific Frequency Computations in accordance with a preferred embodiment of the present invention.

## DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

The Quality of Service (QoS) part of the WipLL system that enables integrated services is generally divided into three major parts:

1. Network access filtering / forwarding;
2. QoS server; and
3. Air access classified queuing.

The network access part is amenable for the routing / bridging of the incoming data, in such a way that all traffic passing further into the system is destined for "over the air" addresses. The QoS server is responsible for traffic behavior, shaping, application recognition, and for the classified queuing that executes QoS policies. Fig. 1 illustrates global data flow within the system, from the network and to it.

There are 3 types of traffic in networks: Best Effort traffic; Profiled traffic and On-Demand traffic.

Best Efforts is traffic as is known today. The traffic goes out onto the network and it is hoped that it reaches its destination, as there are no bandwidth controls and no guarantees. In the future, the present invention envisages continuing to apply best efforts traffic characterization to, e.g., email and unimportant web traffic.

Profiled traffic has pre-defined rules/policies applied to it. These policies may include bandwidth limits, priorities, reservations, security and other controls that characterize this as 'special' traffic.

On-Demand traffic that requires new policies to be applied as the associated applications are loaded. An example might be an unscheduled video-conference.

All three traffic types are present in the WipLL system. Bandwidth shaping policies within the QoS Server (QoS-S) should reflect those application traffic types. Discussed below, are the different methods utilized by the QoS-S toward those traffic topologies.

As illustrated in Figure 1, the QoS server (10) is constructed in the following way:

- it identifies (30) whether the incoming packet is a datagram (IP packet). In case it is not, it assigns it a TTL stamp by advising a policy image defined by the user, based on source / destination addresses, packet type (unicast, multicast or broadcast), etc.;

- it analyzes the packet in 4<sup>th</sup> layer (Connection layer, 20), and assigns the relevant policy; and

- it recognizes a packet's generating application and assigns TTL (Time to Live, 70) stamp.

As protocol (3<sup>rd</sup> layer) recognition (30) is straight forward, let us discuss the connection layer analysis. Note: connection layer analysis (20) is performed for IP traffic only, i.e., UDP, ICMP and TCP protocols.

#### **UDP analysis (40)**

The User Datagram Protocol (UDP) makes available a datagram mode of packet-switched computer communication in the environment of an interconnected set of computer networks assuming that the Internet Protocol (IP) is used as the underlying protocol.

UDP provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. UDP is transaction oriented, and does not guarantee delivery and duplicate protection. Applications requiring ordered reliable delivery of streams of data should use the Transmission Control Protocol (TCP). Every application using UDP has to assign a port number for itself. All port numbers are unique (there is a one to one mapping between application and its port number). Applications that operate at a session type connection are obligated to

introduce three port numbers. Assume some host A communicates to host B, via a VOIP application. The application at A has generated a session initialization datagram having a specific port number recognizable to the application at B as a session generation packet. B replies with a session datagram having some other port number – this is already a port number that specifies a session packet of the passive host B (a session port number). At this point, the session is established, and A communicates to B using a third port number, defining the active connection side. When one of the hosts terminates the session, a session closure datagram containing the first port number is sent.

UDP has three major disabilities: it does not carry any information regarding flow or congestion control, Session Packets reordering is not tolerated and no connection layer correction mechanism is present.

When a UDP datagram is identified, the QoS-S operates in the following way:

- it identifies the application by using its port number;
- it checks whether this packet belongs to an already open session by comparing port numbers and session's participant's IP addresses;

- in an open session, it stamps the packet with a TTL from the applications lookup table;

- in a new session, it advises with the policy agent to determine whether this session is allowed to initiate. The policy agent utilizes the following items: network load (air), median air latency, application's required bandwidth, and hosts access rights.

- it informs the MAC on the application's covenant, in terms of CS air MAC-address; and

- it informs the MAC about session end events (session end, or session failure).

### **ICMP analysis (50)**

The IP is used for host-to-host datagram service in a system of interconnected networks called the Catenet. The network connecting devices are called Gateways. These gateways communicate between themselves for control purposes via a Gateway to Gateway Protocol (GGP).

Occasionally a gateway or destination host will communicate with a source host, for example, to report an error in datagram processing using the Internet Control Message Protocol (ICMP). ICM, uses the basic support of IP as if it were a higher level protocol, however, ICMP is actually an integral part of IP, and must be implemented by every IP module.

ICMP messages are sent in several situations e.g., when a datagram cannot reach its destination, when the gateway does not have the buffering capacity to forward a datagram, and when the gateway can direct the host to send traffic on a shorter route. IP is not designed to be absolutely reliable. The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable. There are still no guarantees that a datagram will be delivered or a control message will be returned. Some datagrams may still be undelivered without any report of their loss. Higher level protocols that use IP (UDP or TCP) must implement their own reliability procedures if reliable communication is required.

ICMP messages typically report errors in the processing of datagrams. To avoid the infinite regress of messages about messages etc., no ICMP messages are sent about ICMP messages. Also ICMP messages are only sent about errors in handling fragment zero of fragmented datagrams. (Fragment zero has the fragment offset equal zero).

In the ICMP QoS-S operational mode:

the user assigns the ICMP packet with default TTL (Time-to-Live). The system default setting is equivalent to the shortest TTL possible – any communication between a gateway and a host is time critical, as most ICMP traffic is generated when queue congestion occurs within a gateway (or beyond it);

the access system, i.e., the MAC protocol delivers this packet (ICMP packet loss can lead to temporal queue distillation within the gateway, even for running applications); and

QoS-S checks the depth of the access queue. Long queues (long is defined in terms of median access latency within the queue) are petrified to lower queue's exhaustion. The QoS-S rejects new sessions (those sessions that are allowed to be rejected are defined by the user) for that relaxation period. However, before

rejecting sessions, flow control can be implemented for TCP/IP traffic. In most cases this effects the required queue relaxation.

### TCP analysis (60)

TCP provides connection-oriented services for the protocol's application layer i.e. the client and the server must establish a connection to exchange data. TCP transmits data in segments encased in datagrams, along with checksums used to detect data corruption, and sequence numbers to ensure an ordered byte stream. TCP is considered to be a reliable transport mechanism because it requires the receiving computer to acknowledge not only the receipt of data but its completeness and sequence. If the sending computer does not receive notification from the receiving computer within an expected time frame, the segment is retransmitted. TCP also maintains a flow control window to restrict transmissions. The receiver advertises a window size, indicating how many bytes it can handle.

TCP provides the following reliability checks:

- acknowledges receipt of packets;
- retransmits when dropped packets are detected;
- re-sequences segments, if necessary, if they arrive out of order;
- tosses packets if data became corrupt during transmission;
- discards duplicate segments; and
- maintains flow control to manage a connection's transmission rate.

### The Bandwidth challenge

TCP/IP was primarily designed to support two traffic applications - FTP and Telnet. With the growth of the Internet, network applications and user expectations have changed. Today, with more high-speed users, and bursty, interactive Web traffic, greater demands are placed on networks, causing delays and bottlenecks that impact a user's quality of service. Many of the features that make TCP reliable, including retransmitting when the network "cloud" drops packets or delays acknowledgment, and backing off when it infers congestion exists, contribute to performance problems.

Conventional TCP bandwidth management uses indirect feedback to infer network congestion. TCP increases a connection's transmission rate until it senses a

problem and then it backs off. It interprets dropped packets as a sign of congestion. The goal of TCP is for individual connections to burst on demand to use all available bandwidth, while at the same time reacting conservatively to inferred problems in order to alleviate congestion.

TCP uses a sliding window flow-control mechanism to increase the throughput over wide-area networks. It allows the sender to transmit multiple packets before it stops and waits for an acknowledgment. This leads to faster data transfer, since the sender doesn't have to wait for an acknowledgment each time a packet is sent.

The sender "fills the pipe" and then waits for an acknowledgment before sending more data. The receiver not only acknowledges that it received the data, but it advertises its window size i.e. how much data it can now handle.

TCP's slow-start method attempts to alleviate the problem of multiple packets filling up router queues. TCP flow control is typically handled by the receiver, which tells the sender how much data it can handle. The slow-start method, on the other hand, uses a congestion window, which is a flow-control mechanism managed by the sender. With TCP slow-start, when a connection opens only one packet is sent until an ACK is received. For each received ACK, the congestion window increases by one. For each round trip, the number of outstanding segments doubles, until a threshold is reached. In summary, TCP uses flow control, determined by client and server operating system configurations, distances, and other network conditions. QoS-S provides rate control, explicitly configured in user-defined policies.

### **Bandwidth management approaches**

When faced with bandwidth constraints, a number of solutions are available including:

Using Queuing Schemes on Routers:

Class queuing: and

Defining Precise Control - The QoS-S Solution.

### **Queuing Schemes on B/Routers**

For the most part, network devices have kept pace with evolving high-speed technology. Routers provide queuing schemes e.g. WFQ, priority output queuing, and

custom queuing that attempt to prioritize and distribute bandwidth to individual data flows so that low-volume applications, such as interactive Web applications, don't get overtaken by large data transfers, typical of FTP traffic.

B/Router-based queuing schemes have several limitations:

B/Routers manage bandwidth passively, tossing packets and providing no direct feedback to end systems;

B/Routers can only use queuing--that is, buffering and adding delay--or packet tossing, to try to control traffic sources;

B/Router queuing is uni-directional--outbound traffic only;

Queuing results in chunkier traffic and erratic performance because multiple, independent TCP sources compete for bandwidth, ramping up and backing off, and queues accumulate at the access link. Queuing, especially WFQ does not work well for chunky flows because packets arriving in chunks tend to be discarded;

B/Routers don't allow setting guaranteed rates for specific traffic type; and

B/Routers can't prevent "brown-outs"--that is, they don't provide admissions-control policies to dictate what happens when a link is over-subscribed.

### Class Queuing

As the access mechanism of the WipLL system differs from the network 2<sup>nd</sup> layer (Ethernet), queuing is inevitable. In order to maintain 4<sup>th</sup> layer decisions (within the QoS server), ordered "licking" is expected from the queues. This is done by assigning an interface header between the 4<sup>th</sup> and the 2<sup>nd</sup> layers for each arriving packet. This header contains information regarding packet's TTL and transmission policy (described later).

### Defining Precise Control - The QoS-S Solution

Traffic, by nature, consists of chunks of data that accumulate when multiple independent sources of data are combined. These data chunks tend to form at access links where speed conversion is handled.

Imagine putting fine sand, rather than gravel, through a network pipe. Sand can pass through the pipe more evenly and quickly than chunks. QoS-S conditions

traffic so that it becomes more like sand than gravel. These smoothly controlled connections are much less likely to incur packet loss and more importantly, the end user experiences consistent service.

Where TCP relies on indirect network feedback from tossed packets to infer congestion, QoS-S provides direct feedback to the transmitter by detecting a remote user's access speed and network latency and correlating this data with aggregate flow information. This results in smoothed traffic flow.

#### **How QoS-S works - Rate Control vs. Flow Control**

The QoS-S maintains state information about individual TCP connections, giving it the ability to provide direct, quality-of-service feedback to the transmitter. In addition, the user can define QoS-S policies to explicitly manage different traffic classes and partition bandwidth resources to meet his business needs. As a result precise control of service levels is gained. QoS-S, as described in this invention, provides several key functions that differentiate it from other bandwidth-management solutions:

- it controls the end-to-end connection, eliminating burstiness, so users experience smooth, even data displays;
- it classifies traffic for precise control (QoS-S classifies by a specific application) and encapsulates a QoS header; and
- it allocates bandwidth according to user-defined policies.

#### **How QoS-S TCP rate control works**

TCP rate control is very similar in concept to the "just-in-time" product flow control used in manufacturing plants. TCP rate control performs the following steps as shown in Fig. 2:

measures current, instantaneous end-to-end latency (done within the datagram) to know how long it will take for a packet to arrive once we "place an order" (360);

computes when the packets will be needed in order to meet latency bounds (latency in terms of flow control factor, rather than time-sensitivity factor) and rate guarantee (370);

specifies (within packet TCP header) how much data to "order", by setting the TCP window size (380); and

places the "order" in the appropriate time so that the data will arrive just when the other session side expects it, e.g. releases ACK (390).

### Controls of the End-to-End Connection

QoS-S uses two methods to control the rate of TCP transmissions:

it detects real-time flow speed and then delays acknowledgments going back to the transmitter; and

it modifies the advertised window in the packets sent to the transmitter.

QoS-S changes the end-to-end TCP semantics from the middle of the connection. It computes the round-trip time (RTT), intercepts the acknowledgment, and holds onto it for amount of time that is required to smooth the traffic flow without incurring retransmission (RTO). It also supplies a window size that helps the sender determine when to send the packet. This rate-control mechanism is illustrated in Fig. 2 and in the following flow example.

### A QoS-S Data-Flow Example

Figure 3 shows how QoS-S (10) intervenes and paces the data transmission to deliver predictable service. The following steps trace the data transfer shown in figure 2:

A data segment (150) is sent from the sender (140) to the receiver (130).

The receiver acknowledges receipt and advertises an 8000-byte window size (160).

QoS-S intercepts the ACK and determines that the data must be more evenly transmitted otherwise subsequent data segments will queue up and packets will be delayed because insufficient bandwidth is available, as defined by this flow's policy.

QoS-S sends an ACK (170) to the sender, computed to arrive at the sender to cause the sender to immediately emit data, i.e., ACK sequence number plus the window size, which allows the sender to transmit an additional packet (180). Then

the QoS-S sends another ACK (190) to the sender which allows the sender to send packet (200) to the receiver without congestion. Thus Smooth Traffic Flow is achieved with QoS-S.

Without the benefit of QoS-RDC, multiple packets are sent; an intermediate router queues packets; and when the queue reaches its capacity, the router tosses packets, which must re-transmitted. Figure 4 A+B shows bursty traffic (210.1...210.7) when QoS-S is not used, and even data transmission (220.1...220.7) under the control of QoS-S.

However, independent of access-link congestion problems, traffic chunks are more prone to loss packet than evenly spaced traffic.

#### **Classifies Traffic for Precise Control**

QoS-S uses a hierarchical tree structure to classify traffic. The user identifies the traffic types to be controlled, such as traffic from a particular application. The user need not classify all network traffic, only the traffic requiring QoS. QoS-S classifies a traffic flow by traversing the traffic tree, attempting to match the flow to one of the classes defined by the user. The final step in the classification process maps a flow to a policy which defines the type of service this traffic class has to receive e.g. a guaranteed rate.

The QoS-S traffic classification function :

- provides a classification for specific applications;
- maintains a traffic class hierarchy to manage priorities and enables policy inheritance; and
- orders traffic classes automatically (by TTL, for the queuing phase).

#### **Controlling Admissions**

The user defines what happens if a traffic class's total guaranteed rate is used up. If the next connection for a class needs a guaranteed rate and no bandwidth is available, QoS-S can handle the bandwidth request by either refusing the connection, or squeezing the connection into the existing bandwidth pipe.

## Scaling Bandwidth to Connection Speed for Efficient Bandwidth use

QoS-S monitors a connection's speed and adjusts bandwidth allocation as the speed changes. Low-speed connections and high-speed connections can be assigned guaranteed rates so that QoS-S can scale bandwidth usage accordingly. For example, during a typical Web session, the wait period between clicks doesn't consume bandwidth, QoS-S frees up this unused but otherwise unavailable bandwidth to satisfy other demands, such as TBS (VOIP, MPEG, etc.).

## Prioritizing Bandwidth Allocation

Priority-based policies are preferred for traffic that doesn't require a reserved guaranteed rate, but are still preferably managed along with competing traffic. The user assigns a priority (0-255) to a traffic class so that QoS-S can determine how to manage the aggregate flow. The user doesn't have to classify all traffic. Any traffic that was not classified is treated as priority-based traffic with a "default priority".

## QoS-S Bandwidth Allocation Order

QoS-S uses the policies defined to determine how to allocate bandwidth. When determining bandwidth allocation, QoS-S takes into account all bandwidth demands, not just the individual traffic flows.

## Classified Queuing (90)

All packets are stamped with a TTL (70) parameter. Stations are expected to order all packets for transmission in TTL ascending order, sending the lowest TTL first.

The basic approach is that (logically) packets arriving (along with their time-to-live stamp) to the transmit-queue cluster, are assigned to the queue according to their time-to-live indicator. As applications are consistent in assigning time-to-live stamps to their outgoing packets, packets from the same application will not be reordered. The reordering will occur between applications, and this is a do-not-care case.

On a continuous basis, as a background task, packets are expected to be deleted from the queue if their updated time-to-live become zero, with one exception -

TCP datagrams. As explained above, TCP datagrams contain session control information, and losing this data will lead the application losing bandwidth. Thus the QoS-S should indicate to the MAC queuing handler whether or not the packet should be discarded upon TTL vanishing.

The MAC (air access) coordinator located in the base station (default gateway of the wireless segment) grants remote units (EPUs) access to the air domain. The MAC coordinator allocates its remote units basing on their "urgency" to transmit. This "urgency" parameter is a combined factor of queue length and TTL distribution within it, computed by each remote unit.

Classified queuing satisfies two system goals:

station queue priority resolution, i.e., prioritization between applications generated over the same remote unit; and

normalization of channel starvation magnitude between remote units for optimal channel time allocation.

Thus as described hereinabove, the QoS server is responsible for network and application layers policies execution including analyzing each incoming packet, detecting its session, evaluating channel load, performing flow control operations (such as delaying packets, intervening into the connection layer, etc.) and attaching a QoS header (over the air) to a packet describing the packet's boundary conditions (such as retransmission criteria, TTL, etc.), as illustrated in Fig.5.

It is appreciated that the software components of the present invention may, if desired, be implemented in ROM (read-only memory) form. The software components may, generally, be implemented in hardware, if desired, using conventional techniques.

It is appreciated that the particular embodiment described in the Appendices is intended only to provide an extremely detailed disclosure of the present invention and is not intended to be limiting.

It is appreciated that various features of the invention which are, for clarity, described in the contexts of separate embodiments may also be provided in combination in a single embodiment. Conversely, various features of the invention

which are, for brevity, described in the context of a single embodiment may also be provided separately or in any suitable subcombination.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention is defined only by the claims that follow:

CLAIMS

1. A quality of service system including:
  - a congestion avoidance subunit; and
  - a traffic flow control unit.
2. Quality of service server apparatus comprising:
  - a protocol detector; and
  - a connection layer analyzer comprising:
    - a UDP analyzer;
    - a TCP analyzer; and
    - an ICMP analyzer.
3. Server apparatus according to claim 2 wherein the UDP analyzer comprises a rate controlled UDP analyzer.
4. Server apparatus according to claim 2 wherein the TCP analyzer comprises a rate controlled TCP analyzer.
5. Apparatus according to claim 2 wherein the UDP analyzer is operative to perform at least some of the following steps:
  - identify application by using its port number;
  - check whether packet belongs to already open session by comparing port numbers and session's participant's IP addresses;
  - if it is an open session, stamp packet with TTL from applications lookup table;
  - if it is a new session, consult with policy agent to determine whether this session is allowed to initiate;
  - inform MAC on application's covenant, in terms of CS air MAC-address; and
  - inform MAC about session end events.

6. Apparatus according to claim 2 wherein the TCP and/or performs at least some of the following reliability checks:

- acknowledges receipt of packets;
- retransmits when dropped packets are detected;
- re-sequences segments, if necessary, if they arrive out of order;
- tosses packets if data became corrupt during transmission;
- discards duplicate segments; and
- maintains flow control to manage a connection's transmission rate.

7. A system according to claim 1 wherein the congestion avoidance unit is operative to perform classified queuing.

8. A system according to claim 1 wherein the rate of TCP transmissions is at least partly controlled by detecting real-time flow speed and then delaying ACKs going back to the transmitter.

9. A system according to claim 1 wherein the rate of TCP transmissions is at least partly controlled by modifying the advertised window size in the packets sent to the transmitter.

8. A system according to claim 7 wherein the classified queuing performed by the congestion avoidance unit comprises assigning packets, arriving with a time-to-live stamp to the transmit-queue cluster, to a queue according to their time-to-live indicator.

9. A quality of service system comprising:

- an adaptive network filtration and forwarding agent;
- a quality of service server; and
- a classified queuing mechanism.

10. A system according to claim 9 wherein said agent is operative to forward packets that belong to the wireless channel only while filtering out irrelevant data streams and competing over the air channel.
11. A system according to claim 9 wherein the quality of service server is operative to execute network and application layers policies including executing at least one of the following: analyzing each incoming packet to detect its session, evaluating channel load, performing flow control operations such as delaying packets and intervening into the connection layer, and attaching a quality of service header to the packet describing the packet's boundary conditions.
12. A system according to any of the preceding claims wherein rate control is carried out such that queues within the access system are kept at substantially a constant length, thereby to reduce session jitter.
13. A system according to claim 9 wherein the classified queuing mechanism provides TTL adaptive access latency within the MAC domain, thereby to enable improved channel bandwidth control for queued data regimes.
14. Apparatus according to any of the preceding claims 1-13 and substantially as shown and described above.
15. Apparatus according to any of the preceding claims 1-13 and substantially as illustrated in any of the drawings.

For the Applicant,

Sanford T. Colb & Co.

C:32682

FIG. 1

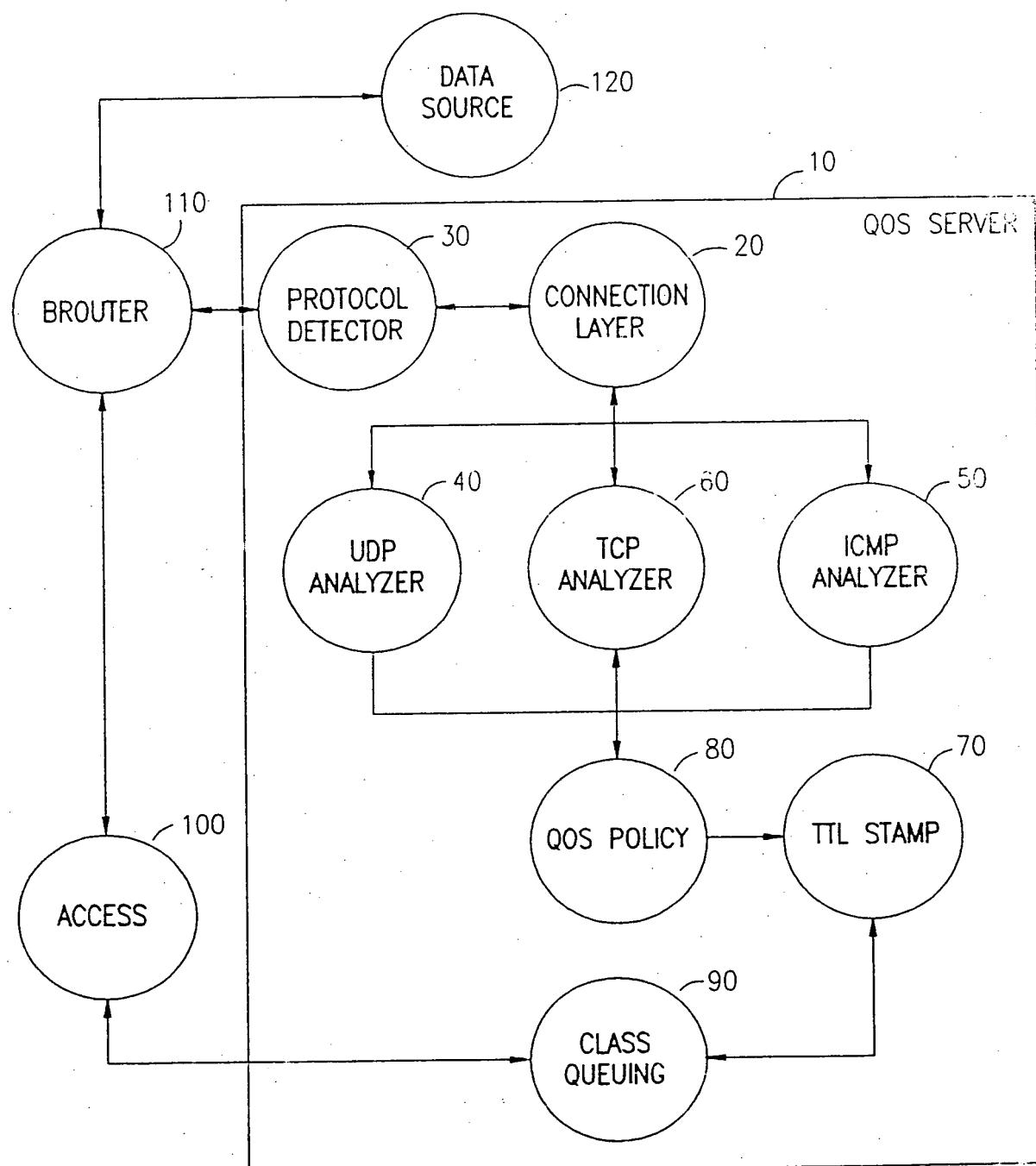


FIG. 2

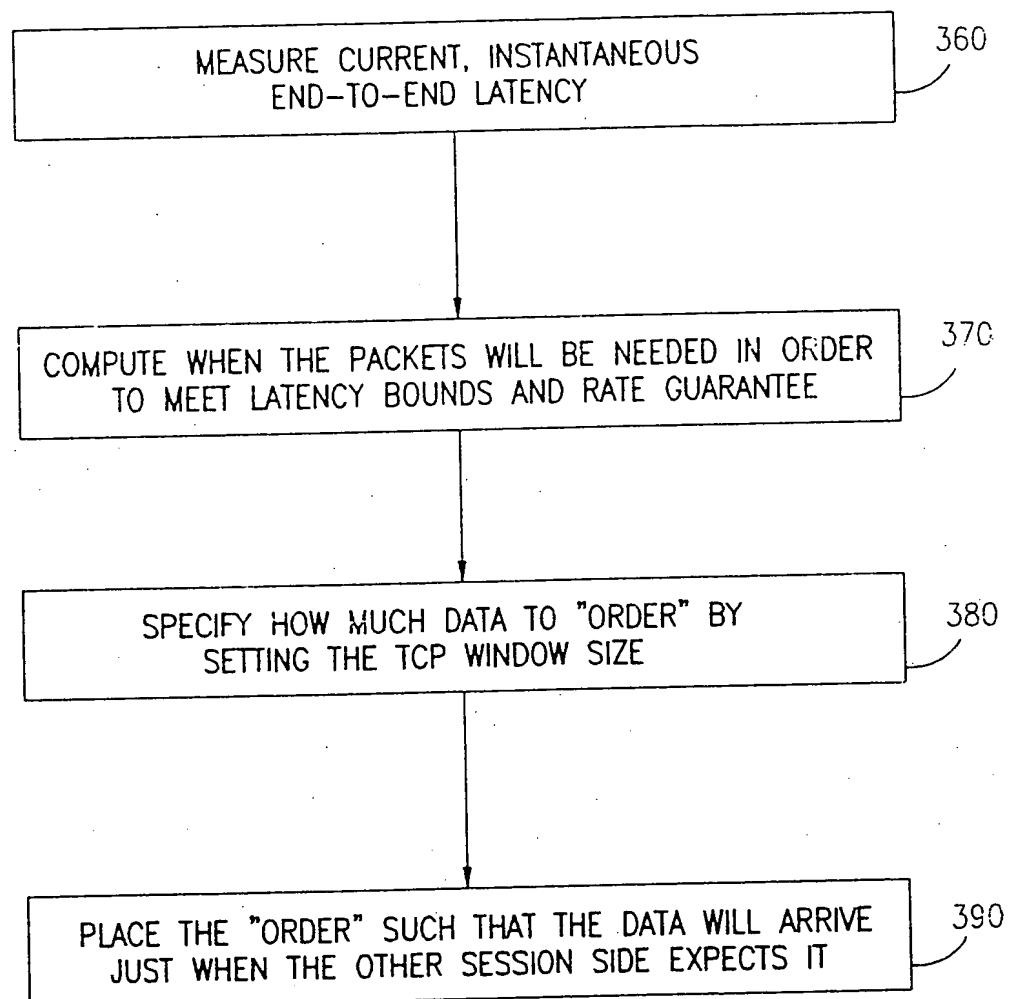


FIG. 3

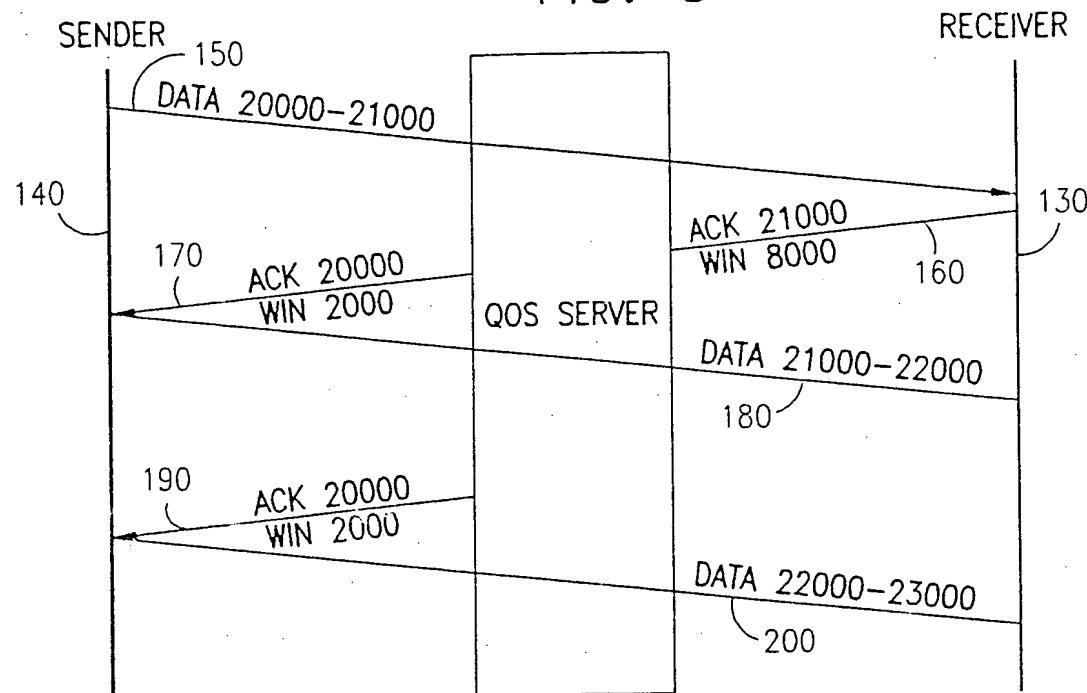


FIG. 4A

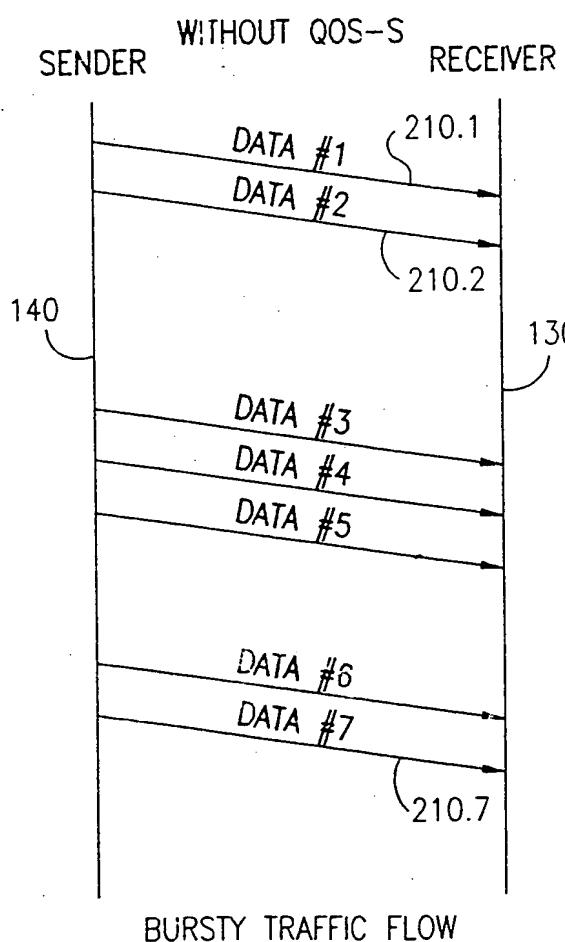


FIG. 4B

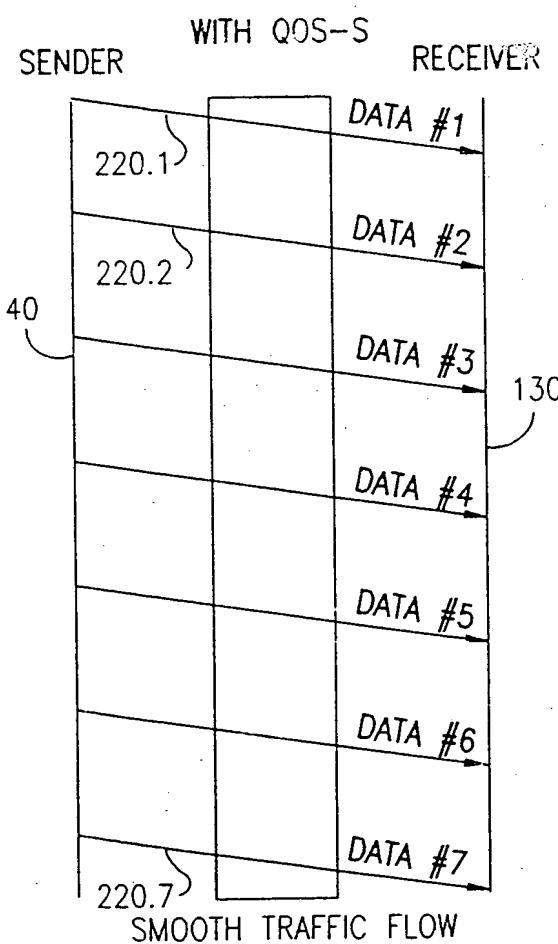
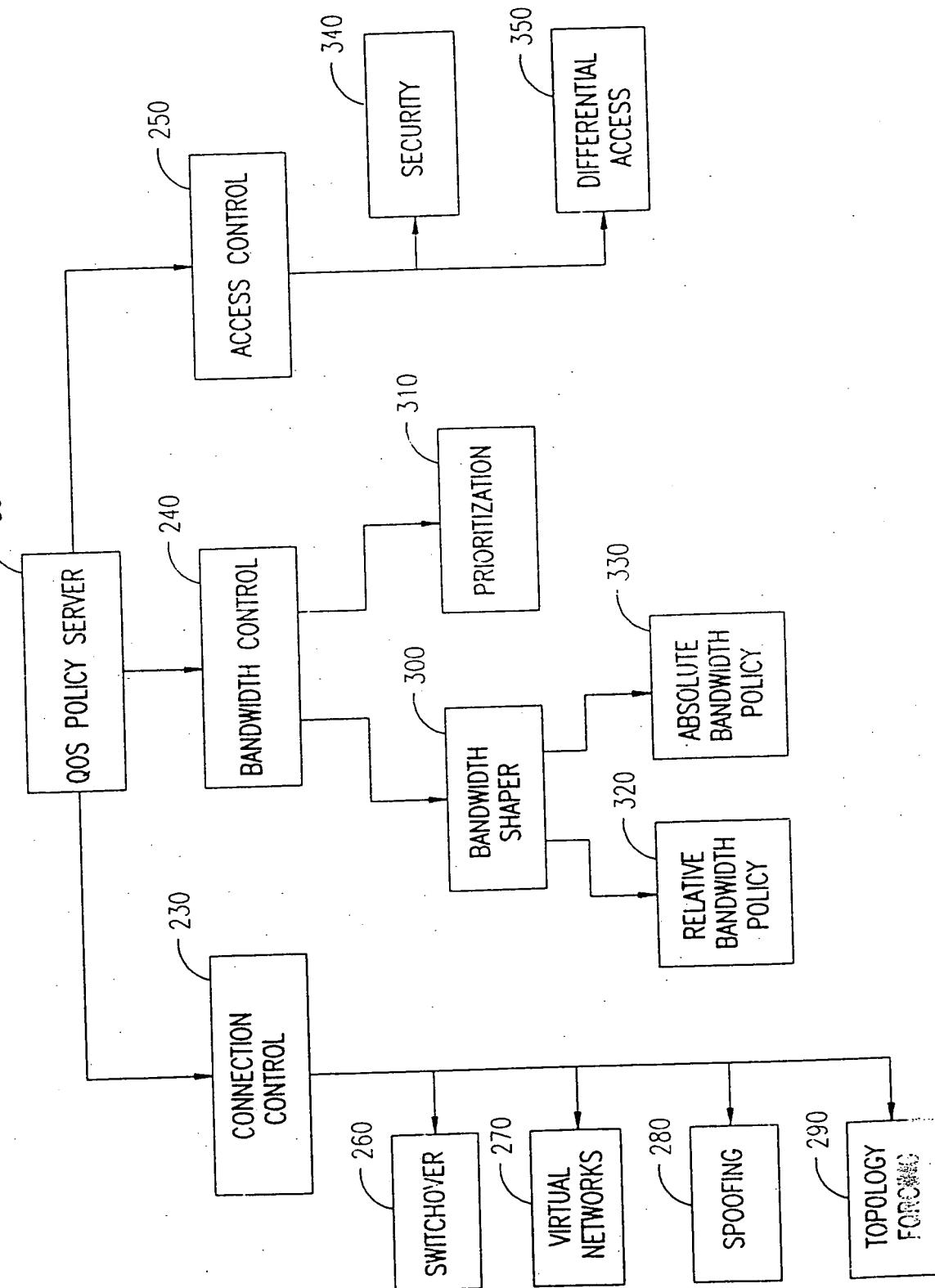


FIG. 5



APPENDIX A

### 1. 1.1.1. Access Parameter Set - Definition

The CC will acquire the following list of parameter set. For each parameter it will be specified how the CC received the parameter and how he updates him.

1) Fragment Error Rate  $fer_j$ .

Notes:

- 1.1 this parameter is to be computed by the CC. See clause (8.3.3).
- 1.2 The CC update  $fer_j$  after each polling.

2) The Basic Channel Assignment  $bch_j$ .

Notes:

- 2.1 The Network manager assigns this parameter.

3) The channel usage  $b_j$ .

Notes:

- 3.1 This parameter computed by the CC, see clause (8.3.5).
- 3.2 The CC updates  $b_j$  after each Polling.

4) Time passed from the last Poll-Tx time :

$$(1.3) \quad t_0 - t_{last}$$

Where  $t_0$  is the current Time.

Notes:

- 4.1 This parameter computed by the CC.
- 4.2 The CC will update this parameter after each Polling.

5) Time To Live of the HOL packets in the station's queue:  $TL_j^1$ .

Notes:

- 5.1 The Time to stamp delivered in the CS Poll\_ACK and in the PRD see (8.4.3).
- 5.2 The CC will update  $TL_j^1$  after each Polling.

6) Packet length of the HOL packet in the station's queue:  $Plength_j^1$

Notes:

- 6.1 The HOL Packet lengths delivered in the CS Poll\_ACK and in the PRD see (8.4.3).
- 6.2 The CC will update  $Plength_j^1$  after polling the j station.

7) Queue weighted average of the TLs:  $\langle TL \rangle_j$ :

$$(1.4) \quad \langle TL \rangle_j := \sum_{i=2}^{Q_j} (aMaxTL - TL_j^i) * \max(1, \sum_{i=2}^{Q_j} \left\lceil \frac{TL_{CR}}{TL_j^i} \right\rceil) * Qlength_j$$

Where:  $TL_{CR}$  stands for a critical Time to Live and is management MIB item.  
aMaxTL stand for the maximum Time To Live stamp allowed and is also a MIB item.

$Qlength_j$  Is the number of packets in the queue of station j

Notes:

- 7.1 The queue weighted average computed by the CS's.
- 7.2 The CS's will update  $\langle TL \rangle_j$  as a background task.
- 7.3 Because the  $\langle TL \rangle_j$  parameter could be very large number the CS will not delivered him in exact manner. The number that the CS send to the CC in the CS Status field is called basic specific frequency.(see next paragraph)

8) The Basic specific frequency  $basicf_j$ :

This parameter represents the urgency of the station to transmit data, the parameter is actually the Queue weighted average writing in 16 bits field.

The operator, that translate  $\langle TL \rangle_j$  to the  $basicf_j$ , is described in section 8.4.3.

Notes:

- 8.1 The  $basicf_j$  will be delivered in the CS Status field of the CS *PHY Header*.
- 8.2 The CC will update  $basicf_j$  after Polling the j station

9) The specific frequency  $f_j$ .

The specific frequency defined as the relative urgency for a station to transmit, normalized to its basic channel usage.

The specific frequency computed by the CC according to the following formula.

$$(1.5) f_j = \left( \frac{1}{Max(1, b_j)} \right) * basicf_j$$

When:  $b_j$  Number of transferred bits in sliding window see Eq (1.2).

Notes:

- 9.1 This parameter computed by the CC.
- 9.2 The CC will update  $f_j$  according to clause 8.4.3.

10) The CS's shall be divided by the CC to three functional categories:

Associated and Active: if all the CS parameters are differ then null and the CS is Associated.

Association and non-Active: if all the CS parameters are nulled and the CS is Associated.

Not Associated: If the CS is not responding for *NRP* (Not Responded Polling) times the CC will consider this CS as Non associated.